



# Détection et suivi basé modèle pour des applications spatiales

Antoine Petit, Eric Marchand, Keyvan Kanani

## ► To cite this version:

Antoine Petit, Eric Marchand, Keyvan Kanani. Détection et suivi basé modèle pour des applications spatiales. Congrès francophone des jeunes chercheurs en vision par ordinateur, ORASIS'13, Jun 2013, Cluny, France. pp.1-6. hal-00829827

**HAL Id: hal-00829827**

**<https://hal.science/hal-00829827>**

Submitted on 3 Jun 2013

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Détection et suivi basé modèle pour des applications spatiales

M. Antoine Petit<sup>1</sup>

M. Eric Marchand<sup>2</sup>

M. Keyvan Kanani<sup>3</sup>

<sup>1</sup> INRIA, Rennes, France

<sup>2</sup> IRISA, Rennes, France

<sup>3</sup> Astrium, Toulouse, France

<sup>1</sup>antoine.guillaume.petit@inria.fr, <sup>2</sup>eric.marchand@irisa.fr, <sup>3</sup>keyvan.kanani@eads.astrium.net

## Résumé

*Cet article concerne la problématique de navigation dans le cadre de rendez-vous spatial, autonome, non coopératif avec des cibles de type satellites, véhicules ou débris spatiaux. Afin de localiser complètement la cible par rapport à un véhicule dit chasseur à l'aide d'un capteur de vision, une technique de détection et de suivi basé modèle est proposée. Notre approche de suivi traite ainsi le modèle 3D complet d'objets présentant une géométrie complexe, de tout type de forme, en tirant profit de l'accélération matérielle. A partir du modèle rendu, des correspondances sont établies avec les contours de l'image et la tâche d'estimation de pose est assurée par une minimisation non linéaire. Pour la détection, qui permet d'initialiser le suivi, le calcul de la pose est basée sur une segmentation premier plan/second plan et sur une procédure efficace de matching d'une série d'images initiales avec des vues synthétiques. Nos méthodes ont été évaluées sur des images synthétiques et réelles.*

## Mots Clef

Suivi 3D basé modèle, détection, reconnaissance 3D, robotique spatiale.

## Abstract

*This paper focuses on navigation issues for space autonomous, uncooperative rendezvous with targets such as satellites, space vehicles or debris. In order to fully localize, using a vision sensor, a chaser spacecraft with respect to a target spacecraft or debris, a visual model-based detection and tracking technique is proposed. Our tracking approach processes complete 3D models of complex objects, of any shape by taking advantage of GPU acceleration. From the rendered model, correspondences are found with image edges and the pose estimation task is then addressed as a nonlinear minimization. For detection, which initializes the tracking, pose estimation is based on foreground/background segmentation and on an efficient contour matching procedure with synthetic views, over a few initial images. Our methods have been evaluated on both synthetic images and real images.*

## Keywords

Model-based tracking, 3D detection, space robotics.

## 1 Introduction

La désorbitation de débris spatiaux lourds (au delà d'une tonne) a été identifiée comme un enjeu majeur afin de contrôler la prolifération de ces débris et de limiter le risque pour des satellites actifs. En ce sens, l'Agence Spatiale Européenne (ESA) a récemment lancé des projets comme le Geostationary Servicing Vehicle (GSV) ou le RObotic GEostationary orbit Restorer (ROGER), tous deux destinés à capturer, inspecter, assister ou désorbiter des satellites défaillants. Actuellement, Astrium élabore également des solutions de navigation, au niveau capteur, à bord d'un véhicule d'évacuation de débris, avec comme principal objectif d'assurer un haut niveau de fiabilité lors de manoeuvres de proximité. Ici, nous nous concentrons sur des techniques permettant la détection et le suivi 3D d'une cible complexe, en se basant sur son modèle 3D et en utilisant une caméra monoculaire, dans le but d'estimer l'état relatif entre le chasseur et sa cible.

Notre approche a été testée sur des séquences d'images réelles, comme le rendez-vous du Soyuz-TMA avec l'ISS ou la manoeuvre d'approche de la navette Atlantis, pour valider sa robustesse à des données réelles. Des tests sur des images synthétiques ont aussi été menés pour montrer quantitativement les performances et la robustesse du processus de suivi.

### 1.1 Suivi basé modèle 3D

Les approches courantes de suivi basé modèle utilisent soit une information de type point d'intérêt [1], soit de type contour [5, 3] ou une combinaison des deux [14]. Les contours offrent une bonne invariance aux changements d'illumination ou au bruit et sont particulièrement adaptées pour des scènes ou objets faiblement texturés. Pour ce type d'approche, le calcul de pose est réalisé en minimisant la distance entre les contours projetés du modèles 3D et les contours correspondants dans l'image, extraits en recherchant le maximum de gradient le long des normales aux contours. Des techniques d'optimisation numérique non-

linéaire pondérée sont utilisées pour la minimisation. Afin de rejeter des erreurs ou aberrations, des méthodes comme RANSAC [1] ou des M-estimateurs [14, 3] sont des solutions courantes pour rendre l'algorithme robuste aux occultations ou aux changements d'illumination. La plupart de ces techniques traitent des modèles 3D constitués de lignes. Mais réaliser la projection du modèle complet dans l'image conduit à certaines limites pour des objets présentant des formes cylindriques, sphériques, courbes, complexes. Des modèles polygonaux complets pour ce type d'objet s'avèrent trop lourds et nécessitent ainsi d'être remodelés manuellement afin de conserver les contours les plus significatifs de la scène et d'obtenir un temps de calcul acceptable. Un premier enjeu pour notre approche est donc de pouvoir traiter un modèle polygonal complet. Ceci donc afin de s'affranchir de cette phase de remodelage et d'être en mesure d'exploiter toute l'information contenue dans le modèle. Notre méthode repose ainsi sur l'utilisation du processeur graphique et d'un moteur de rendu 3D, afin d'opérer la projection du modèle automatiquement et de déterminer les contours visibles et saillants directement à partir de la scène rendue. Une méthode de ce type a également été adoptée par [15]. Un avantage de cette technique est de pouvoir gérer automatiquement les faces cachées ou les auto-occultations. Un deuxième enjeu est l'amélioration de la robustesse en combinant les contours de profondeur et avec des contours de texture et en intégrant une approche multi-hypothèses dans le processus d'appariement des contours.

## 1.2 Détection basée modèle

Comme le suivi se réalise image par image, ce processus nécessite d'être initialisé. Dans la littérature, de nombreuses méthodes de reconnaissance reviennent à apprendre et classifier des sacs de mots 2D extraits de différentes vues de l'objet recherché. Bien qu'elles aient prouvé leur efficacité, ces techniques s'avèrent peu adaptées à notre cas, où on considère des objets 3D faiblement texturés. De plus, travailler sur des images d'apprentissage réelles est peu réalisable dans notre contexte. Pour ces raisons nous proposons de nous baser sur la forme globale et la saillance de l'objet, à partir d'une base de vues synthétiques de l'objet à partir du modèle 3D. Cette méthode de détection, présentant une certaine genericité et s'apprenant d'une manière peu supervisée, peut être lourde en temps de calcul et trop approximative en considérant une simple image [4, 13, 9], et le suivi ne pourrait être par la suite correctement initialisé. C'est pourquoi nous proposons d'étendre le processus sur une séquence de quelques images initiales. Cela permettrait d'une part de suivre l'objet sur cette séquence, tout en raffinant sa pose et en maintenant un temps de calcul acceptable. En ce sens, ces travaux se rapprochent de ceux proposés par [12].

Le principe général est d'apparier de manière robuste chaque image avec une vue synthétique type du modèle, conjointement à une position, une orientation et une taille

dans l'image, paramètres qui sont estimés par filtrage partiel. Un estimé initial grossier de ces paramètres est nécessaire et est évalué grâce à une segmentation de l'objet au premier plan par rapport à l'arrière plan.

Le reste de l'article est organisé de la manière suivante. La section 2 donne un aperçu de la méthode proposée pour le suivi et la section 3 de celle pour la détection. La section 4 présente quelques résultats expérimentaux.

## 2 Suivi basé modèle 3D

### 2.1 Approches classiques et problématique

Notre problème concerne le suivi basé modèle, à partir d'un modèle 3D de la cible. Le but est d'estimer la pose de la caméra qui permet le meilleur alignement entre les contours du modèle projeté et les contours extraits de l'image. De telles approches ont prouvé leur efficacité et divers auteurs ont proposé des formulations différentes du problème [5, 3]. L'idée principale est la suivante. Etant donné une nouvelle image, le modèle 3D de la scène ou de l'objet est projeté dans l'image suivant la pose estimée précédente  $\mathbf{r}$ . Chaque ligne projetée  $l_i(\mathbf{r}) = pr(L_i, \mathbf{r})$  du modèle est ensuite échantillonnée, générant un ensemble de points 2D  $\{\mathbf{x}_i\}$ . Puis à partir de chaque point  $\mathbf{x}_i$  on procède à une recherche monodimensionnelle le long de la normale au contour projeté afin de trouver un point correspondant  $\mathbf{x}'_i$  dans l'image. Pour calculer la nouvelle pose, les distances entre les points  $\mathbf{x}'_i$  et les lignes projetées  $l_i$  sont minimisées suivant le critère suivant [3] :

$$\Delta = \sum_i \rho(d_{\perp}(l_i(\mathbf{r}), \mathbf{x}'_i)) \quad (1)$$

où  $d_{\perp}(l_i(\mathbf{r}), \mathbf{x}'_i)$  est la distance entre un point  $\mathbf{x}'_i$  et la ligne correspondante  $l_i(\mathbf{r})$ .  $\rho$  est un estimateur robuste, qui réduit la sensibilité de l'algorithme à des aberrations. Ce problème revient à une minimisation non-linéaire par rapport à la pose  $\mathbf{r}$ . Dans [3], ce processus suit la structure d'asservissement visuel virtuel, similaire à une approche de type Gauss-Newton.

L'idée de considérer des cibles à la forme complexe, qu'elles soient texturées ou non, conduit à oublier la notion de ligne 3D comme dans [3] ou dans des travaux antérieurs [8] et de considérer seulement des points 3D appartenant indifféremment à des contours saillants ou à des contours de silhouette ou à des contours de textures. Deux enjeux sont ainsi à considérer : la projection d'un modèle complexe et la sélection de points 3D.

Notre algorithme de suivi est structuré de la manière suivante :

- Projection du modèle complet par rapport à la pose  $\mathbf{r}_k$  estimée pour l'image précédente  $I_k$ . Cette étape est réalisée rapidement grâce au processeur graphique par l'intermédiaire d'OpenGL, même pour un modèle présentant une grande complexité.
- À partir du modèle projeté, des points de mesure 3D  $\mathbf{X}_i$  sont générés en extrayant les contours provenant de discontinuités de profondeur ou de texture.

- Recherche d'un point de contour correspondant  $\mathbf{x}'_i$  dans l'image  $I_{k+1}$ . Afin d'améliorer la robustesse, nous proposons un version multi-hypothèses de l'algorithme.
- La dernière étape consiste à estimer la pose  $\mathbf{r}_{k+1}$  qui minimise les erreurs  $d(\mathbf{x}_i, \mathbf{x}'_i)$  entre les points  $\mathbf{x}'_i$  extraits de l'image et la projection des points de mesure 3D sélectionnés  $\mathbf{x}_i(\mathbf{r}) = pr(\mathbf{X}_i, \mathbf{r})$ , avec le critère suivant :

$$\Delta = \sum_i \rho(d(\mathbf{x}_i(\mathbf{r}), \mathbf{x}'_i)) \quad (2)$$

## 2.2 Génération de points de mesure 3D

À chaque image acquise  $I_{k+1}$ , le modèle est rendu et projeté en utilisant un moteur de rendu 3D se basant sur OpenGL, selon la pose précédente  $\mathbf{r}_k$ . Notre but est d'obtenir un ensemble de points 3D  $\mathbf{X}_i$  qui appartiennent aux contours de silhouette, aux contours saillants ou de texture à partir des textures et de la carte de profondeur rendues. Notre approche suit ainsi celle de [15] et se rapproche des techniques de génération de silhouette de modèles polygonaux décrites dans [7].

**Extraction de contours en utilisant la profondeur et la texture.** À partir de la carte de profondeur, qui correspond aux valeurs de profondeur de la scène selon la position de la camera, pour chaque pixel (Figure 1(a)), nous pouvons déterminer les discontinuités qui correspondent à l'apparence géométrique de la scène. C'est pourquoi, en appliquant un filtre Laplacien du second ordre à ces valeurs de profondeur, il en résulte une carte binaire de contours visibles de la scène ((Figure 1(b)). L'étape de filtrage et d'extraction de contours est ici implémentée sur carte graphique en utilisant la programmation par *shader*, afin d'accélérer le calcul.

Pour des scènes ou des objets présentant des textures, les contours géométriques sont parfois insuffisants et des ambiguïtés avec des contours de textures peuvent apparaître, provoquant de mauvais appariements et ainsi des minima locaux au cours de l'estimation de pose. Une amélioration apportée par notre méthode consiste ainsi à combiner ces deux types de contours. Les textures rendues du modèle 3D sont ainsi traitées par un détecteur de Canny et les contours obtenus sont ajoutés à ceux générés à partir de la carte de profondeur.

**Génération de points de mesure 3D.** Etant donnée la carte de contour de la scène complète, les coordonnées 3D des points de contour dans la scène peuvent être calculés grâce à la carte de profondeur et à la pose utilisée pour projeter le modèle. Comme traiter toute la carte de contour peut être coûteux en temps de calcul, nous échantillons selon les coordonnées  $x$  et  $y$  du plan image afin de garder un nombre raisonnable de points. Par ailleurs, la phase de suivi (voir section 2.3) nécessite de connaître l'orientation du contour sous-jacent au point de mesure  $\mathbf{x}_i$ . Pour les contours de texture, ceci est réalisé à partir de l'algorithme de Canny sur les textures rendues. Pour les contours de profondeur, les gradients de Sobel sont calculés selon

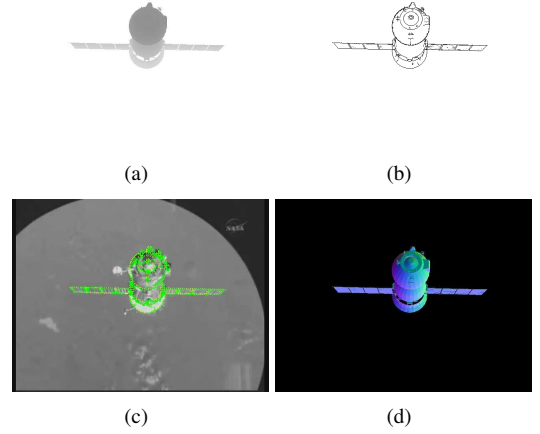


FIG. 1: Sur (a) est représenté la carte de profondeur du modèle 3D rendu, à partir duquel la carte de contour est calculée (b). Cette carte de contour est ensuite échantillonnée pour extraire des points de mesure, reprojetés sur l'image courante et à partir une recherche monodimensionnelle le long de la normale au contour est réalisée pour trouver un point de contour correspondant dans l'image acquise (c). (d) montre la carte des normales de la scène.

$x$  et  $y$  sur le niveau de gris de la carte des normales de la scène. La carte des normales associe la valeur [R,G,B] de chaque pixel aux coordonnées dans l'espace de la normale à la surface correspondante (voir Figure 1(d)). Ces étapes basiques de traitement d'image et de rendu sont aussi effectuées sur le processeur graphique.

## 2.3 Estimation de pose et suivi bas-niveau

**Suivi à partir des points de mesure.** Les points de mesure sont ensuite traités afin d'extraire des contours correspondants dans l'image. A la manière de [14, 3, 15], nous procédons à une recherche monodimensionnelle le long de la normale au contour sous-jacent (Figure 2 et Figure 1(c)) de chaque  $\mathbf{x}_i$ . Une approche classique est de choisir le pixel présentant le maximum de gradient comme point d'appariement  $\mathbf{x}'_i$  dans l'image.

**Minimisation d'une distance à une ligne.** Une fois que les correspondances sont établies, le but est ensuite d'estimer la nouvelle pose  $\mathbf{r}_{k+1}$  qui permet un réaligement des points de mesure issus du modèle avec les points de l'image  $\mathbf{x}'_i$ . Cette étape est assurée en minimisant les erreurs  $d(\mathbf{x}_i(\mathbf{r}), \mathbf{x}'_i)$ . D'une manière semblable à [5, 3, 15], notre approche considère la distance entre la ligne 3D projetée  $l_i(\mathbf{r})$  sous-jacente au point de mesure 3D  $\mathbf{X}_i(\mathbf{r})$  et le point d'appariement sélectionné  $\mathbf{x}'_i$  dans l'image (voir Figure 2). Le critère (2) devient :

$$\Delta = \sum_i \rho(d_{\perp}(l_i(\mathbf{r}), \mathbf{x}'_i)) \quad (3)$$

où  $\rho$  est un estimateur robuste utilisé pour rejeter des aberrations (estimateur de Tukey) et  $d_{\perp}(l_i(\mathbf{r}), \mathbf{x}'_i)$  est la distance entre un point  $\mathbf{x}'_i$  et la ligne correspondante  $l_i(\mathbf{r})$ . Il

faut noter que pour des contours saillants le point 3D  $\mathbf{X}_i(\mathbf{r})$  n'est pas modifié lorsque la pose  $\mathbf{r}$  évolue. Ce n'est plus le cas pour les points  $\mathbf{X}_i(\mathbf{r})$  qui appartiennent aux frontières de silhouette. Néanmoins, puisque le mouvement relatif de la camera/objet entre deux images successives est supposé faible, cette approximation n'a pas d'effet sur l'efficacité de l'approche.

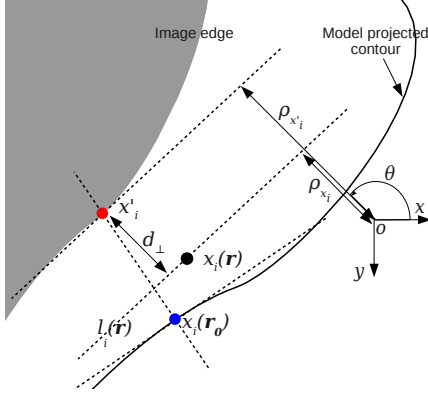


FIG. 2: À partir de la pose initiale  $\mathbf{r}_0$ , une recherche monodimensionnelle est réalisée le long de la normale au contour projeté sous-jacent au point de mesure  $\mathbf{x}_i(\mathbf{r}_0)$ . Minimization de la distance  $d_{\perp}$  entre le point  $\mathbf{x}'_i$  et la ligne  $l_i(\mathbf{r})$ .

La technique d'optimisation est ensuite similaire à l'asservissement visuel virtuel décrit dans [3] et qui assimile ce problème d'optimisation à un problème d'asservissement visuel.

**Solution multi-hypothèses.** Afin d'améliorer la robustesse de l'estimation de pose et d'éviter des problèmes dus aux ambiguïtés entre contours, il est possible de considérer différentes hypothèses correspondant à des contours potentiels. Ils correspondent à plusieurs maxima locaux du gradient le long de la normale. Comme dans [14, 11], nous choisissons le point candidat qui a la distance la plus courte avec la ligne 3D projetée  $l_i$  au cours du processus de minimisation. La fonction de coût devient :

$$\Delta = \sum_i \rho(\min_j d_{\perp}(l_i(\mathbf{r}), \mathbf{x}'_{i,j})) \quad (4)$$

où les points  $\mathbf{x}'_{i,j}$  sont les candidats sélectionnés pour chaque point de mesure  $\mathbf{x}_i$ .

### 3 Détection basée modèle

Dans cette partie est abordée le problème d'initialiser notre algorithme de suivi, pour un objet 3D faiblement texturé, avec un arrière plan potentiellement bruité. La solution retenue repose seulement sur le modèle 3D géométrique de l'objet. L'estimation de pose par détection est réalisée itérativement sur une séquence d'images initiales, d'une manière proche de [12], en profitant de l'hypothèse que l'on traite un objet cible en mouvement par rapport au fond. Un aperçu de la méthode est décrit ci-après (voir Figure 3) :

- **Etape d'apprentissage hors ligne** destinée à construire, à partir d'une base de vues synthétiques, un graphe de vues hiérarchique résultant en des vues types  $V$  du modèle.
- **Etape de détection en ligne** :
  - Segmentation de la silhouette de l'objet en utilisant une technique de segmentation premier plan/second plan. En calculant les moments binaires de la silhouette extraites, un estimé initial de son barycentre  $(x_c, y_c)$ , son orientation  $\theta$  et son aire  $A$  peut être déterminé.
  - Pour chaque vue type, un filtrage particulière sur la variable  $\mathbf{x} = [x_c \ y_c \ \theta \ A]^T$  est établi.
  - Grâce à une structure probabiliste, à chaque image, la vue type la plus probable et un estimé correspondant  $\hat{\mathbf{x}}$  sont déterminés, donnant ainsi la pose complète.
  - Lorsque qu'une vue type atteint une probabilité suffisante par rapport aux autres, le processus est arrêté. La pose est enfin raffinée en parcourant la structure hiérarchique à partir de la vue sélectionnée.

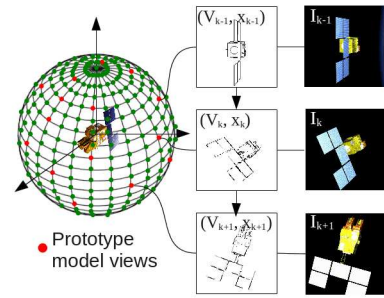


FIG. 3: Principe du processus de détection. Des vues synthétiques sont générées autour d'une sphère de vues. Les transitions entre les vues sélectionnées suivent un modèle probabiliste type Champ Aléatoire Conditionnel.

#### 3.1 Graphe de vues hiérarchique

**Generation de vues synthétiques.** Le principe de la méthode est d'associer à chaque image de la séquence initiale une vue synthétique générée à partir du modèle 3D, selon un critère de similarité. Ces vues synthétiques doivent couvrir l'espace des poses de dimension six. Nous proposons, comme dans [4, 13, 9, 12], de générer ces vues sur une sphère de vues centrée sur le modèle 3D, paramétrée par deux degrés de liberté (latitude et longitude). Ceci est réalisé en plaçant des caméras virtuelles uniformément réparties sur la sphère à l'aide d'un moteur de rendu 3D (Figure 3). A partir des vues rendues, les contours de profondeurs sont extraits par la technique décrite en section 2.2. Ainsi, à la différence de [12], nous ne considérons à la fois les contours de silhouette et les contours intérieurs de l'objet.

**Etablir un graphe de vues hiérarchique.** Puisque ce processus d'appariement vue synthétique/image peut être coûteux en temps de calcul si on considère toute la base de vues synthétiques, ces dernières sont itérativement classifiées pour former un graphe de vue hiérarchique. Au premier niveau de cette structure, des classes de vues sont établies par Propagation d’Affinité [6] sur des régions disjointes de la sphère de vue, afin d’alléger la charge mémoire. Ceci est réalisé en comparant les vues les unes avec les autres selon un critère de similarité basé contours de type Chamfer Matching. Le résultat est un ensemble de classes, chacune représentée par une vue type. Nous procédons de cette manière avec ces vues types obtenues. Ainsi nous pouvons construire itérativement des niveaux hiérarchiques successifs jusqu’à atteindre un nombre raisonnable de vues types (entre 50 et 100 dans notre cas, à partir d’une base de 1000 vues).

### 3.2 Segmentation et extraction de silhouette

Comme l’appariement des vues types du modèle avec les images en entrée permet d’obtenir deux degrés de liberté (la latitude et la longitude autour de l’objet), une solution est proposée pour estimer les quatre paramètres de silhouette que sont la position, l’orientation et la taille de l’objet dans l’image. Elle repose sur une segmentation de la silhouette de l’objet, qui est supposé être en mouvement dans l’image par rapport à l’arrière plan. Nous avons élaboré une technique dans laquelle les déplacements de l’objet et de l’arrière plan dans l’image sont modélisés, en classifiant des trajectoires de points d’intérêt suivis par l’algorithme Kanade-Lucas-Tomasi (KLT), en faisant l’approximation d’un fond planaire et stationnaire pour un fond consistant en la surface terrestre. La classification des trajectoires de points est réalisée par la méthode décrite dans [10]. La probabilité d’appartenance de chaque pixel à l’objet ou au fond se mesure ainsi selon cette modélisation du objet/fond en terme de couleur. De plus, en supposant le mouvement du fond plan, une technique de compensation de mouvement basé homographie est aussi utilisée dans le calcul des probabilités. Enfin ces probabilités sont intégrées dans une fonction d’énergie minimisée efficacement par graphcut [2], permettant d’attribuer l’appartenance de chaque pixel à l’objet en premier plan ou à l’arrière plan.

### 3.3 Un modèle probabiliste pour associer les images initiales avec les vues types

Cependant les paramètres de silhouette, déterminés en calculant les moments binaires de la silhouette segmentée, sont parfois évalués de manière trop grossière à cause d’erreurs de segmentation. Il est ainsi nécessaire de les raffiner afin de progressivement trouver une vue type cohérente. Une méthode par filtrage particulière sur ces paramètres pour chaque vue type est ainsi adoptée en ce sens. La vraisemblance de chaque particule pour chaque vue se base là-aussi sur un critère de type Chamfer. Afin d’assurer des transitions lisses entre les vues type déterminées d’une

image à une autre, les probabilités des vues sont déterminées suivant un Champ Aléatoire Conditionnel.

## 4 Résultats

Le processus de rendu du modèle 3D polygonal repose sur la librairie OGRE, sur-couche d’OpenGL. Comme précisé auparavant, certaines étapes de traitement d’image sont implémentées sur processeur graphique en utilisant des shaders sous le langage GLSL. En terme de hardware, une carte graphique NVIDIA NVS 3100M a été utilisée, avec un processeur 2.8GHz Intel Core i7.

Des tests ont été réalisés à la fois sur des images réelles et sur des images de synthèse. Ils ont permis de valider l’algorithme de suivi présenté, ainsi que la technique de détection.

### 4.1 Tests sur des images réelles

Le premier exemple traite du suivi du vaisseau Soyuz TMA-12 lors de son rendezvous avec la Station Spatiale Internationale (ISS). Le vaisseau présente des formes complexes, bien que les panneaux solaires permettent de faciliter le suivi. Ici le processus de suivi consiste à seulement utiliser les contours de profondeurs (voir Figure 1(d)), et la solution à hypothèse unique sur le suivi bas niveau s’est avérée suffisante pour cette séquence. Malgré quelques incertitudes concernant les paramètres internes de caméra, la fidélité du modèle et la basse qualité de la vidéo, la phase de suivi se réalise correctement (voir Figures 5(a)- 5(d)). Pour la détection, nous observons que bien que la segmentation ne soit pas très précise (Figure 4(b)), cinq images initiales (trois sont montrées ici, Figures 4(b)-4(d)) permettent d’aboutir à une vue type suffisamment probable avec des paramètres d’image (position, orientation, taille) acceptables pour initialiser le suivi.

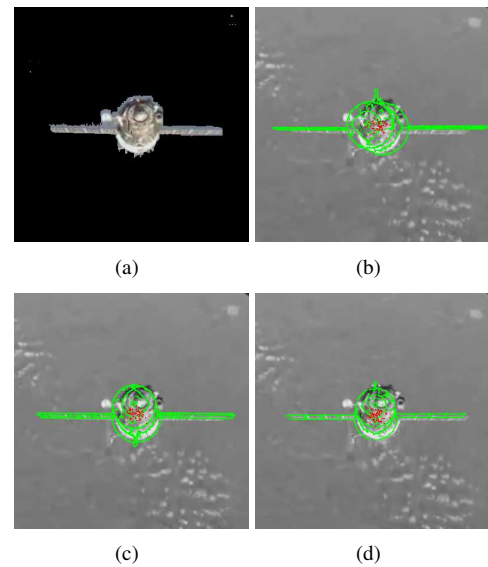


FIG. 4: Détection pour la séquence Soyuz. A partir de la segmentation (a) le processus finit par converger vers une vue acceptable (d).



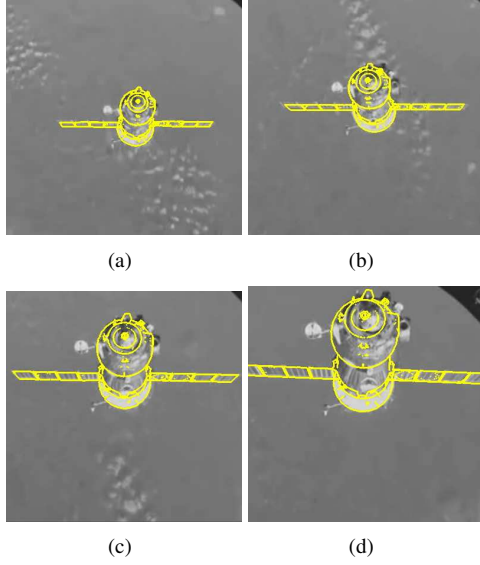


FIG. 5: Suivi, à partir des contours de profondeur, pour la séquence Soyuz.

Le second exemple concerne la navette spatiale Atlantis réalisant une manoeuvre lors de son approche de l'ISS. Un modèle 3D non texturé de la navette a été traité pour le suivi, et la solution multi-hypothèses, ainsi qu'un filtre de Kalman à vitesse constante sur les paramètres de pose, ont été utilisés. La Figure 6 montre les résultats pour la phase de détection. La segmentation (Figure 6(a)) permet d'extraire relativement finement la silhouette de la navette, d'où une pose correctement déterminée après trois images initiales ((Figures 6(b), 6(c), 6(d)) pour ensuite initialiser le suivi. Les Figures 7(a)- 7(f) montrent le suivi s'effectuant de manière satisfaisante sur la séquence, avec une certaine robustesse aux changements d'illumination. La solution multi-hypothèses ainsi que le filtre de Kalman ont été nécessaires pour gérer correctement la rotation de la navette (Figure 7(d)).

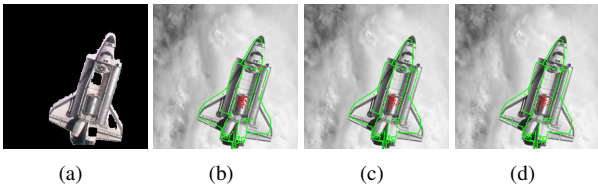


FIG. 6: Détection pour la séquence Atlantis.

## 4.2 Tests sur des images synthétiques

Ces tests ont nécessité l'utilisation d'un simulateur d'images en ray-tracing développé par Astrium pour des environnements spatiaux. Nous nous concentrons ici sur le cas de la famille des satellites Spot. Dans un contexte de débris spatial, nous considérons une cible présentant un mouvement de rotation quelconque et un vaisseau chasseur est supposé être situé sur une orbite similaire, avec une excentricité légèrement différente afin qu'il puisse survoler la

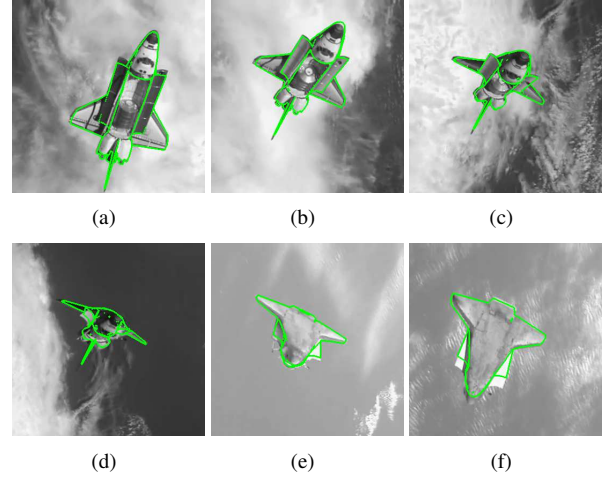


FIG. 7: Suivi reposant sur des contours de profondeur pour la séquence Atlantis, avec à la solution multi-hypothèses, en utilisant un filtre de Kalman.

cible, dans le plan  $x_{Orb} - z_{Orb}$  du repère orbital (voir Figure 8).

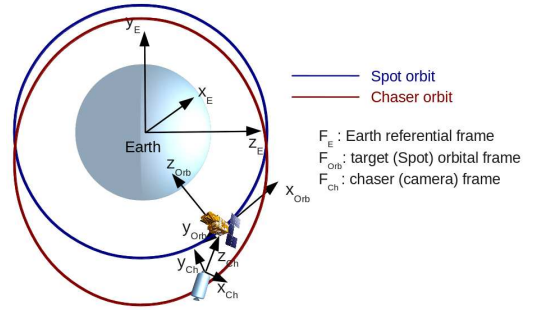


FIG. 8: Orbites du chasseur et la cible (spot) dans le repère lié à la Terre.

Comme cela peut être observé sur les Figures 9(f)- 9(m), le suivi présente de bonnes performances tout au long de la séquence. Afin de quantifier les différents résultats, nous avons séparément évalué la précision en rotation et en translation d'une pose estimée de la caméra  $\hat{r}$  par rapport à la pose vraie  $r^*$ , sachant qu'une vérité terrain est disponible.

Les résultats pour les approches simple et multi hypothèse(s) (SH et MH), qui reposent à la fois sur des contours de profondeur et de texture, et avec ou sans filtre de Kalman, sont représentées sur la Figure 10. Ils montrent qu'en utilisant la solution multi-hypothèse avec un filtre de Kalman, les erreurs en translation et rotation peuvent être maintenues assez faibles, particulièrement lorsque la cible est loin du chasseur avec une faible luminosité (see Figure 9(l)), et lorsque les panneaux solaires basculent dans l'image (voir Figure 9(j)), conduisant à un minimum local pour la solution simple hypothèse. Les Figures 9(b)-9(e) montrent le processus de détection qui initialisent correc-

tement le suivi, malgré une segmentation assez grossière (Figure 9(a)).

### 4.3 Temps de calcul

Grâce à l'implémentation de plusieurs phases sur la carte graphique, le temps d'exécution a pu être considérablement réduit par rapport à une approche exclusivement sur CPU. L'algorithme de suivi peut ainsi être traité à environ 15 images/s. La solution multi-hypothèses affecte peu le temps de calcul et rajouter les contours de textures est plus coûteux, puisque plus de points sont considérés, avec une fréquence de 7 image/s pour la séquence Spot. La technique de détection tourne elle à 0.5 image/s.

## 5 Conclusion

Cet article présente des méthodes de détection et de suivi appliquées à des objets complexes, potentiellement texturés, dans un environnement spatial, pour des applications de rendez-vous spatial et de désorbitation de débris. Le suivi repose sur la génération de contours saillants à partir d'un modèle polygonal 3D complet, projeté grâce à un moteur de rendu. L'information fournie à la fois par les discontinuités géométriques et de textures de l'objet est exploitée. Cette technique permet d'éviter une phase lourde de remodelage du modèle 3D. La méthode de suivi bas niveau et d'estimation de pose est ensuite similaire à des approches classiques. Une approche de détection également basée modèle a aussi été mise en place pour initialiser le suivi, par l'intermédiaire d'un modèle probabiliste d'appariement entre une séquence d'images initiales et des vues synthétiques de l'objet. Par les tests effectués sur des images réelles, notre approche présente des résultats qualitatifs satisfaisants, à la fois pour le suivi et la détection, tout en présentant des temps de calcul acceptables pour les applications envisagées. Grâce au simulateur réaliste d'images de synthèse, les performances ont pu être mesurées et comparées, montrant plus quantitativement l'intérêt des différentes méthodes.

## Références

- [1] G. Bleser, Y. Pastarmov, and D. Stricker. Real-time 3d camera tracking for industrial augmented reality applications. *Journal of WSCG*, pages 47–54, 2005.
- [2] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. volume 23, pages 1222–1239, Washington, DC, USA, November 2001. IEEE Computer Society.
- [3] A.I. Comport, E. Marchand, M. Pressigout, and F. Chaumette. Real-time markerless tracking for augmented reality : the virtual visual servoing framework. *IEEE Trans. on Visualization and Computer Graphics*, 12(4) :615–628, July 2006.
- [4] Christopher M. Cyr and Benjamin B. Kimia. A similarity-based aspect-graph approach to 3d object recognition. *International Journal of Computer Vision*, 57 :5–22, 2004.
- [5] T. Drummond and R. Cipolla. Real-time visual tracking of complex structures. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(7) :932–946, July 2002.
- [6] Brendan J. Frey and Delbert Dueck. Clustering by passing messages between data points. *Science*, 315 :972–976, 2007.
- [7] T. Isenberg, B. Freudenberg, S. Schlechtweg, and T. Strothotte. A developer guide to silhouette algorithms for polygonal models. *IEEE Comput. Graph. Appl.*, 23(4) :28–37, 2003.
- [8] A. Petit, E. Marchand, and K. Kanani. Vision-based space autonomous rendezvous : A case study. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'11*, pages 619–624, San Francisco, USA, September 2011.
- [9] Christian Reinbacher, Matthias Ruether, and Horst Bischof. Pose estimation of known objects by efficient silhouette matching. In *20th International Conference on Pattern Recognition (ICPR)*, 2010.
- [10] Y. Sheikh, O. Javed, and T. Kanade. Background subtraction for freely moving cameras. In *ICCV*, pages 1219–1225. IEEE, 2009.
- [11] C. Teulière, E. Marchand, and L. Eck. Using multiple hypothesis in model-based tracking. In *IEEE Int. Conf. on Robotics and Automation, ICRA'10*, pages 4559–4565, Anchorage, Alaska, May 2010.
- [12] A. Toshev, A. Makadia, and K. Daniilidis. Shape-based object recognition in videos using 3d synthetic object models. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 0 :288–295, 2009.
- [13] Markus Ulrich, Christian Wiedemann, and Carsten Steger. Cad-based recognition of 3d objects in monocular images. In *Proceedings of the 2009 IEEE international conference on Robotics and Automation, ICRA'09*, pages 2090–2097, Piscataway, NJ, USA, 2009. IEEE Press.
- [14] L. Vacchetti, V. Lepetit, and P. Fua. Combining edge and texture information for real-time accurate 3d camera tracking. In *ACM/IEEE Int. Symp. on Mixed and Augmented Reality, ISMAR'04*, pages 48–57, Arlington, VA, November 2004.
- [15] H. Wuest and D. Stricker. Tracking of industrial objects by using cad models. *Journal of Virtual Reality and Broadcasting*, 4(1), April 2007.



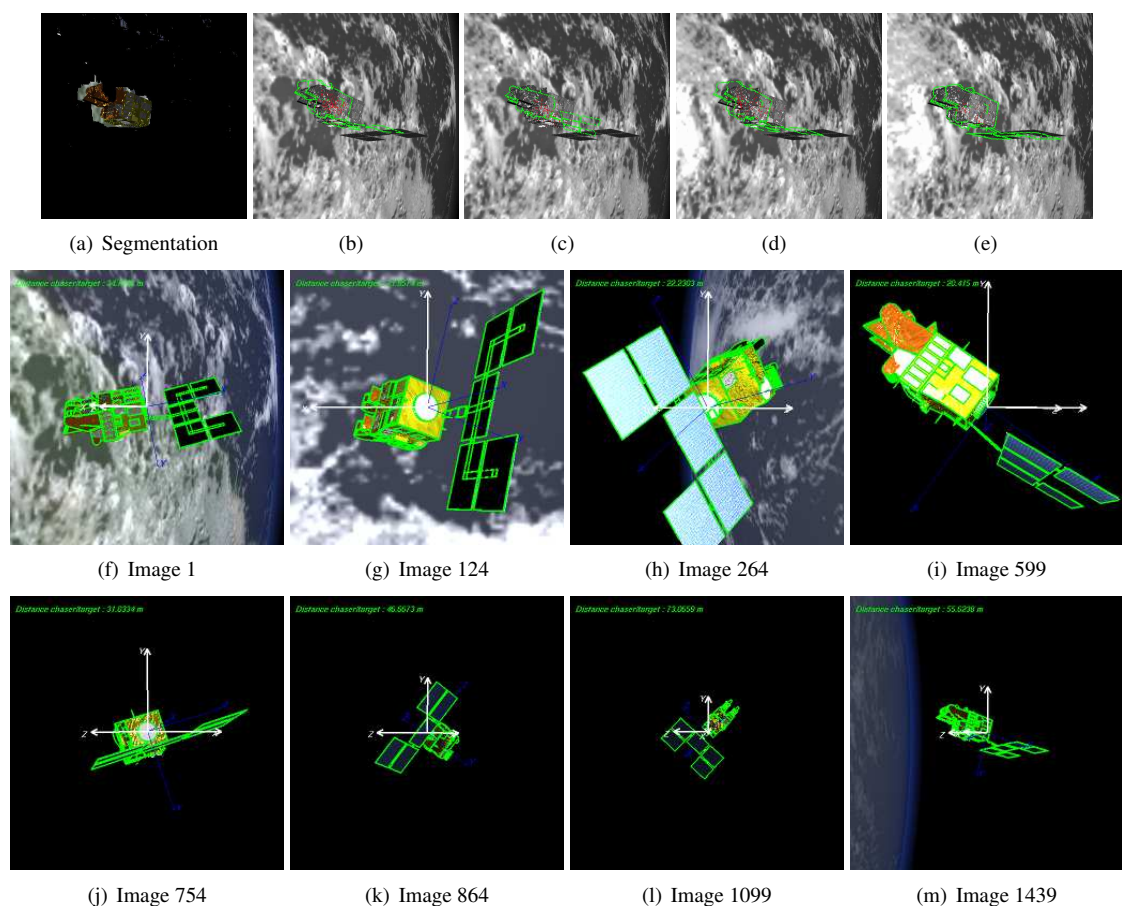


FIG. 9: La segmentation de la cible (a) est assez grossière. La détection met ainsi 8 images avant de converger vers une solution acceptable (4 sont montrées ici, (b)-(e))). Le suivi ((f)-(m)) repose lui sur les contours de profondeur et de texture, avec la solution multi-hypothèses et un filtre de Kalman.

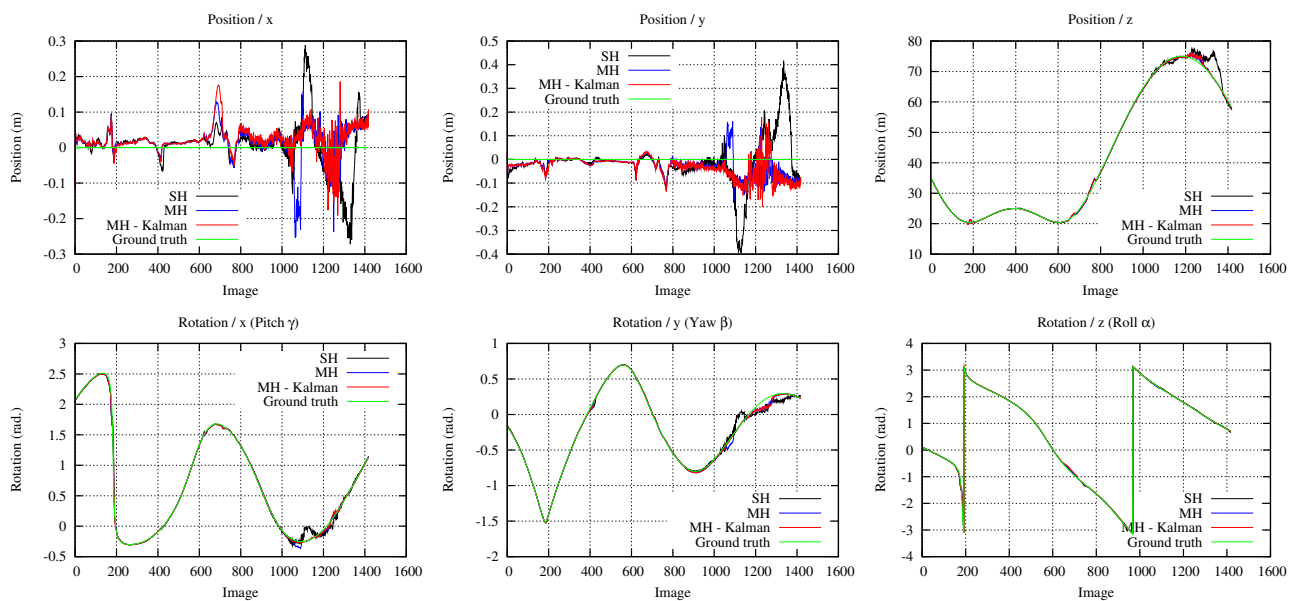


FIG. 10: Pose de la caméra estimée le long de la séquence, avec la vérité terrain, pour les solutions simple et multi-hypothèses, et avec un filtre de Kalman ou non.